# Retrieving Information from Predict

This section covers the following topics:

- Retrieving Information from Documentation Data
- Documentation Data Refers to XREF Data and Vice Versa
- Retrieving Information from XREF Data
- Monitoring the Implementation Process
- Using Sets

## Retrieving Information from Documentation Data

The Predict data dictionary system helps to manage complex information processing systems by providing functions that retrieve information on its internal structure and properties of its components. Information can be retrieved online or in batch mode.
Essentially three types of information can be retrieved from the dictionary.

- **Information on Attributes of Individual Objects**
  Most retrieval types report on the attributes of individual objects. Information on any number of objects can be retrieved in one run.
- **Information on the Associations of Objects**
  Retrieving information on the associations of objects means retrieving information on the structure of the information processing system, for example which programs belong to a system, and which files are used by these programs.
- **Information on the Implementation of Objects**
  For example whether an object that is documented in the dictionary is already implemented. Active retrieval functions as well as features of standard retrieval types provide information on the implementation of objects. The section Retrieving Information from XREF Data describes in more detail all options to retrieve information on the implementation of a system and compare this information with documentation data.

### Overview of Retrieval Options

To determine precisely which information is to be retrieved from the dictionary the following types of parameters are used:

- **Selection Criteria**
  Retrieval operations can be executed for single objects or for groups of objects. Selection criteria determine which objects are to be included in a report.
- **Retrieval Type**
  The retrieval type determines the type of information to be retrieved from the dictionary.
- **Output Mode**
  In addition to the retrieval type,an output mode must be specified. The output mode determines how information retrieved from the dictionary is output and whether objects can be selected for further processing.
- **Output Options**
  The selection criteria and the retrieval type determine the information collected by Predict. Output options can then be used to determine which information is actually contained in a report.
- **Retrieval Models**
  The retrieval type Execute retrieval model reports on the structure of an information processing system. It can be specified exactly which part of a metadata model is to be evaluated.

**Where to find Detailed Information**

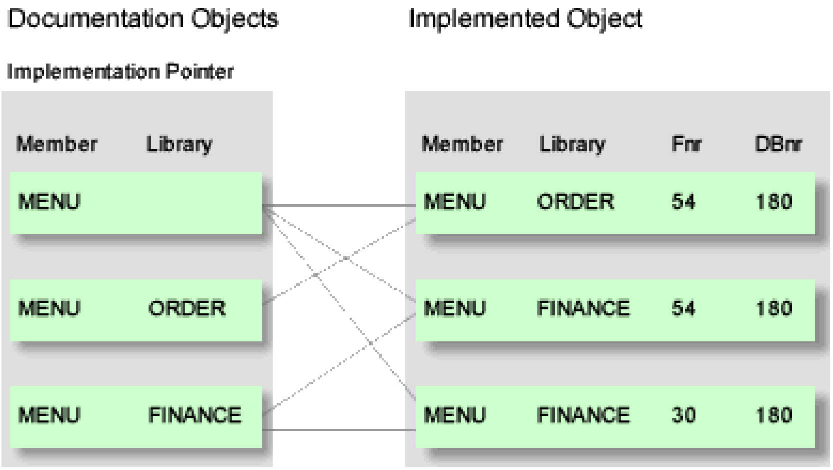
See the section Retrieval in the **Predict Reference documentation**.

# Documentation Data Refers to XREF Data and Vice Versa

Predict retrieval and active retrieval functions evaluate XREF data **and** documentation data, whereby retrieval functions start on the documentation side and active retrieval functions start on the XREF side.

To evaluate (and compare) information on objects stored as documentation data and as XREF data, Predict must know which documentation objects belong to which record in XREF data.
This connection between the documentation and the XREF data is established with the implementation pointer of documentation objects (member name, library name, user system file number and user system database number).

If the same member is used in several libraries, multiple documentation of this member can be avoided by omitting the library name. Predict then finds out for itself all the libraries in which this member exists.



| Member | Library |
|---|---|
| MENU | |
| MENU | ORDER |
| MENU | FINANCE |

| Member | Library | Fnr | DBnr |
|---|---|---|---|
| MENU | ORDER | 54 | 180 |
| MENU | FINANCE | 54 | 180 |
| MENU | FINANCE | 30 | 180 |

## Forcing/Disallowing an Implementation Pointer

The parameters Implementation Library and Implementation DBNR/FNR can be set in the Maintenance Options screen of the General Defaults menu and apply when maintaining program objects.

**F**
Forced: Library and/or DBnr/Fnr must be specified.

**A**
Allowed: Library and/or DBnr/Fnr of the implemented program documented with the Predict program object can be specified.

**D**
Disallowed: Library and/or DBNR/FNR must not be specified.
For third generation languages, library *SYSALL* must be used.

**Where to find Detailed Information**

See the section Using Implementation Pointers to Establish a Connection between Documentation and XREF Data in the section **Active Retrieval** in the **Predict Reference documentation**.

# Retrieving Information from XREF Data

Documentation data alone does not necessarily say much about what has been implemented. It can exist without a single line of code having been written or differ significantly from what has been implemented.
It is the XREF data that completes the picture by documenting what has actually been implemented.

XREF data is of interest to various groups of users: programmers, system analysts, project leaders, database administrators, data dictionary administrators, computer auditors, etc. However, each of these groups requires the XREF data evaluated in different ways. The following two sections contain some hints on how XREF data might be used.

## Evaluating XREF Data for a Specific Application

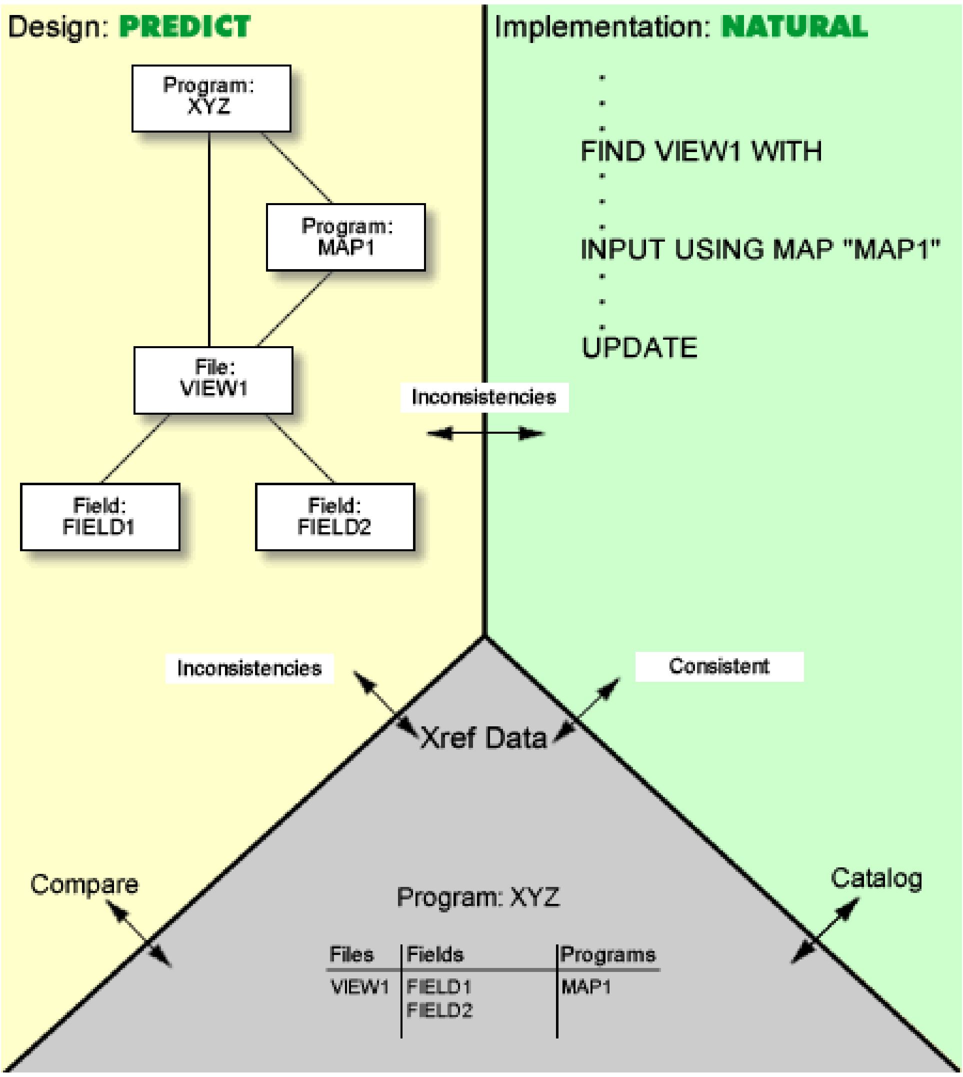
The LIST XREF command provides information on a specific application which can be used to monitor the application. Errors can be detected (unused programs) and opportunities for optimization (reducing the number of variables) are given in early development phases. The impact of program changes can be estimated easily (for example, how many programs must be modified when a field or variable is changed).

The following kinds of evaluation are available:

- How often an object (program, variable, Copy Code, error number or processing rule) is used;
- How a program is used, including recursive invocation structures;
- Which objects are used by a program;
- Which objects are available but not used (for example, which programs are not invoked by any other programs);
- Which files are used, which of their fields are used, and how the fields are used (as counter fields or for reading, storing, searching or updating) in data areas;
- All information about a program (use of programs, files, fields, variables, work files, retained sets, Copy Code, printers and error numbers), the source code of the program and its documentation in the Predict data dictionary.
- Which programs were cataloged when, by whom and from which terminal.

The functions of Natural LIST XREF can be called from a menu or with a command.

# Monitoring the Implementation Process



## Evaluating XREF Data Across Several Applications

Predict active retrieval functions evaluate XREF data across several applications.

When documentation data is evaluated, implementation data can also be output if required. The date when the documentation was last altered shows if changes to programs have been carried out in the documentation. In this way, the state of both the documentation and the implementation of an application can be evaluated.

Comparing the documentation data and the implementation also allows the detection of possible errors, such as descriptor fields that are not used.
Section Comparison in the **External Objects in Predict documentation** describes how documentation and implementation are compared.

According to the type of function, the evaluation can be restricted either by specifying implementation data (member name, library name) or documentation data (object name) as selection criteria.

The following types of evaluation are available:

- Which objects are implemented, including an indication of the documented entries;
- How often an object is used (field, file, external program or verification);
- Which objects are implemented but not documented;
- Which objects are documented but not implemented;

- Which members use documented objects (verifications, files, and fields);
- The files defined as being used by a member and the files that are actually used;
- The programs defined as being used in an application and the members that are actually used.

Using these evaluations, differences between the real implementation of a system and its documentation in Predict can be detected and resolved (see also the diagram above).

## How XREF Data can be Evaluated and where the Different Methods are Described

XREF data can be evaluated on its own or in conjunction with descriptive information stored in data dictionary objects.

## Evaluating XREF Data without Documentation Objects

Two methods are provided for retrieving XREF data independently of information stored in data dictionary objects:

- XREF data written for members in the current Natural library is evaluated with the LIST XREF command. This method is described in the section LIST XREF for Natural in the **Predict Reference documentation**. See also Evaluating XREF Data for a Specific Application.
- XREF data that was generated by Adabas Native SQL, Adabas SQL Server, the Predict preprocessor or the function CREATE DBRM of Natural for DB2 is evaluated with the LIST XREF for 3GL command of Predict. The LIST XREF command for 3GL is described in the section LIST XREF for Third Generation Languages in the **Predict Reference documentation**.

## Evaluating XREF Data Together with Documentation Data

Again, two methods can be used to evaluate XREF data in connection with information stored in Predict documentation objects:

- Predict active retrieval functions compare XREF data with the corresponding information in Predict documentation objects. This method is described in the section Active Retrieval in the **Predict Reference documentation**.
  See also Evaluating XREF Data Across Several Applications.
- With Predict retrieval functions, set the parameter Show implementation in the output options to Y to display the implementation of a documentation object, or set the parameter Mark implementation to Y to mark implemented objects with an asterisk. Information on the implementation is taken from XREF data.

### When is an Object Regarded as Implemented?

What is meant by implemented depends on the object type.

- The following **databases** are regarded as implemented:
  - type A and connected to a physical Adabas C database,
  - type D and connected to a physical DB2 database,
  - type P with a DBnr defined in the NTDB macro as an Entire System Server database,
  - type I.
- A **dataspace** is regarded as implemented if a DB2 tablespace or SQL/DS DBspace has been generated from the Predict dataspace.
- A **file** is regarded as implemented if
  - any kind of copy code has been generated for a file or
  - the file is connected to an external object.
- A **program** is regarded as implemented if XREF data exists.
- A **storagespace** is regarded as implemented if a DB2 storagegroup has been generated from the Predict storagespace.

- A **system** is regarded as implemented if XREF data exists for a program in the library documented by the system.
- A **verification** is regarded as implemented if the rule of the verification is used in a map and XREF data exists.

# Using Sets

Retrieval functions performed on XREF data often produce a list of objects that needs further processing, such as programs to be maintained or cataloged. Predict can save the output list of XREF data retrieval in a set for further processing. Sets can be used to share work among members of a project team and for communicating exactly what work needs to be done.

The following rules apply when working with sets:

- Sets are created by setting the Save set option in a retrieval menu to Y or by using the function Create new sets of LIST XREF.
- Sets can contain Natural objects of all types. Members stored in sets can be cataloged or stowed, their contents can be edited or listed directly from the set.
- Sets can be displayed, purged, sent to another user, merged, subtracted or intersected. When a set is sent, a short comment can be included that will appear when the set is displayed at the terminal of the recipient.
- Sets are saved separately for each user and each Natural library. Any user defined in Predict can create and use up to twelve sets in each library.

**Note:**
Sets can also be used in the Natural utilities NATUNLD and SYSMAIN.

## Example for the Use of Sets

Whenever a Copy Code is changed, all programs that use the Copy Code may need to be changed accordingly. The LIST XREF function Copy code referenced in programs can be used to find out which programs are affected. If the types and names of these programs are saved as a set, a single function can then be used to edit the contents of all the programs, one after another, and change them to match the changed Copy Code.

All members contained in the set can then be recataloged using a single function.

### Where to find Detailed Information

Section LIST XREF for Natural in the **Predict Reference documentation** contains a complete description of handling sets.